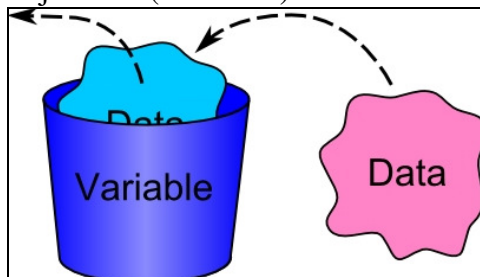


Pripreme 26.09.2015. – Početna grupa (osnovna škola)

1. Deklaracija promjenljivih, tip promjenljive

- a. Promjenljiva je ime za neku memorijsku lokaciju. Sadržaj te lokacije, tj. vrijednost promjenljive se mijenja tokom vremena. Dodjeljivanje nove vrijednosti promjenljivoj uništava njenu postojeću vrijednost. (vidi sliku)



- b. Kada vam treba promjenljiva, morate zatražiti memoriju koja će čuvati vrijednost te promjenljive. Računaru nije potrebna ista količina memorije za npr. jedno slovo i jedan realan broj. Zbog toga morate navesti kakvog je tipa vrijednost koju želite čuvati u promjenljivoj. Dakle, morate navesti tip i ime promjenljive, i moguće zadati njenu početnu vrijednost.
- c. Vodite računa da je skup cijelih brojeva u računaru konačan, za razliku od matematike, gdje je on beskonačan. Takođe, realni (razlomljeni) brojevi se u računaru predstavljaju sa greškom. Realni brojevi se pišu sa decimalnom tačkom umjesto decimalne zapete.
- d. C++ podržava sljedeće osnovne tipove podataka (neke od navedenih tipova obradićemo detaljnije u sljedećim lekcijama):

Naziv	Opis	Veličina	Opseg
char	Karakter – jedan simbol ili mali broj	1 bajt	Sa predznakom: od -128 do 127 Bez predznaka: od 0 do 255
short int ili short	Integer – cio broj	2 bajta	Sa predznakom: od -32768 do 32767 Bez predznaka: od 0 do 65535
int	Integer – cio broj	4 bajta	Sa predznakom: od -2147483648 do 2147483647 Bez predznaka: od 0 do 4294967295
long int ili long	Integer – cio broj	najmanje 4 bajta	
bool	tačno ili netačno		true ili false
float	realan broj (tzv. single precision floating point)	4 bajta	
double	realan broj (tzv. double precision floating point)	8 bajtova	Preciznost najmanje kao float
long double	realan broj	najmanje 8 bajtova	Preciznost najmanje kao double
wchar_t	Tzv. „wide“ karakter	1 bajt	Npr. japansko slovo

- e. Promjenljivu deklariramo tako što navedemo njen tip i njeno ime. Na primjer:

```
int i;  
long n;  
bool indikator;  
float b;  
double visina;
```

- f. Ako želite, možete više promjenljivih definisati odjednom, razdvajajući njihova imena simbolom zapeta:

```
int i, j, k = 32;
long n , n1;
bool indikator, flag, isPrime;
float a, b, c;
double visina, sirina, m123, ugaoRotacije;
```

- g. Pri deklarisanju promjenljive može se zadati i njena početna vrijednost (tzv. inicijalizacija), tako što poslije imena promjenljive napišemo simbol '=' i vrijednost promjenljive:

```
int k = 32;
long n = 0L;
bool indikator = false, flag = true, isPrime;
float a = 3.4f, b, c = 5.1234;
double visina, sirina, m123, ugaoRotacije = 67.88;
```

- h. Postoje još dva načina inicijalizacije promjenljivih. Poslije imena promjenljive u zagradama (velikim ili malim) napiše se njena početna vrijednost. Inicijalizacija pomoću velikih zagrada (tj. simbola '{' i '}') postoji od 2011. godine.

```
int p(10);
int q{20};
```

- i. Primjer programa u kojem se promjenljive inicijalizuju na sva tri načina:

```
// inicijalizacija promjenljivih

#include <iostream>
using namespace std;

int main ()
{
    int a=5;           // pocetna vrijednost: 5
    int b(3);         // pocetna vrijednost: 3
    int c{2};        // pocetna vrijednost: 2
    int rezultat;    // pocetna vrijednost: neodredjena

    a = a + b;
    rezultat = a - c;
    cout << rezultat;

    return 0;
}
```

- j. Imena promjenljivih se zadaju na sljedeći način: prvo slovo imena promjenljive mora biti malo ili veliko slovo engleske abecede, a ostali simboli imena mogu biti slova engleske abecede, cifre i simbol '_' (underscore). Velika i mala slova se razlikuju, pa su npr. ana12, Ana12 i anA12 različita imena promjenljivih. Programerska praksa je da ime promjenljive počinje malim slovom i da imena promjenljivih budu opisna za zadatak koji rješavate. Pogledajte donje tabele za primjere ispravnih i neispravnih imena promjenljivih:

ispravno	Ispravno ali se ne preporučuje
a	A
a1	A1
ugaoRotacije	_ar
ugao_rotacije	DARKO
brojOdlicnihUcenika	Marko_Markovic
marko1	

neispravno	obrazloženje
5marko	Prvo slovo imena je cifra
marko petrovic	Blanko u imenu
Marko#peric	Nije dozvoljen simbol # u imenu
int	Rezervisana riječ jezika C/C++
cout	Rezervisani objekat u jeziku C++

- k. Ključne riječi jezika C++ (sa sajta <http://en.cppreference.com/w/cpp/keyword>). Ove riječi su rezervisane, tj. **ne** smijete ih upotrebiti kao imena promjenljivih:

alignas (since C++11)	else	requires (concepts TS)
alignof (since C++11)	enum	return
and	explicit	short
and_eq	export(1)	signed
asm	extern	sizeof
auto(1)	false	static
bitand	float	static_assert (since C++11)
bitor	for	static_cast
bool	friend	struct
break	goto	switch
case	if	template
catch	inline	this
char	int	thread_local (since C++11)
char16_t (since C++11)	long	throw
char32_t (since C++11)	mutable	true
class	namespace	try
compl	new	typedef
concept (concepts TS)	noexcept (since C++11)	typeid
const	not	typename
constexpr (since C++11)	not_eq	union
const_cast	nullptr (since C++11)	unsigned
continue	operator	using(1)
decltype (since C++11)	or	virtual
default(1)	or_eq	void
delete(1)	private	volatile
do	protected	wchar_t
double	public	while
dynamic_cast	register	xor
	reinterpret_cast	xor_eq

2. Naredba dodjeljivanja

- Opšti oblik naredbe je: `promjenljiva = izraz`
- Prvo se izračuna izraz sa desne strane znaka =, pa se njegova vrijednost dodijeli promjenljivoj sa lijeve strane znaka =.
- Primjer:

```

k = 32;
p = visina * sirina;
m123 = k = 123;
k = j/10 + k%10;
a = (i+j+5.62)*(h-1/2)- a/2;
boolean t = (a+b>c) && (b+c>a) && (c+a>b);

```

- d. Aritmetički operatori: sabiranje (+), oduzimanje (-), množenje (*), dijeljenje (/), moduo – ostatak pri dijeljenju (%)
- e. Relacioni operatori: manje (<), veće (>), manje ili jednako (<=), veće ili jednako (>=), jednako (==) i različito (!=).
- f. Logički veznici: && (konjunkcija, I, AND), || (disjunkcija, ILI, OR), ! (negacija, NE, NOT) – više o ovim veznicima govorićemo na sljedećem času.
- g. Prioritet operatora, tj. redosljed izvršavanja operacija, prikazan je u sljedećoj tabeli (preuzeto sa sajta http://en.cppreference.com/w/cpp/language/operator_precedence). Neke od ovih operatora nismo obradili – govorićemo o njima u sljedećim lekcijama.

Precedence	Operator	Description	Associativity
1	::	Scope resolution	Left-to-right
2	++ --	Suffix/postfix increment and decrement	
	type() type{}	Functional cast	
	()	Function call	
	[]	Subscript	
	. ->	Member access	
3	++ --	Prefix increment and decrement	Right-to-left
	+ -	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(type)	C-style cast	
	*	Indirection (dereference)	
	&	Address-of	
	sizeof	Size-of ^[note 1]	
	new new[]	Dynamic memory allocation	
delete delete[]	Dynamic memory deallocation		
4	.* ->*	Pointer-to-member	Left-to-right
5	* / %	Multiplication, division, and remainder	
6	+ -	Addition and subtraction	
7	<< >>	Bitwise left shift and right shift	
8	< <=	For relational operators < and ≤ respectively	
	> >=	For relational operators > and ≥ respectively	
9	== !=	For relational operators = and ≠ respectively	
10	&	Bitwise AND	
11	^	Bitwise XOR (exclusive or)	
12		Bitwise OR (inclusive or)	
13	&&	Logical AND	

14		Logical OR	
15	?:	Ternary conditional ^[note 2]	Right-to-left
	throw	throw operator	
	=	Direct assignment (provided by default for C++ classes)	
	+= -=	Compound assignment by sum and difference	
	*= /= %=	Compound assignment by product, quotient, and remainder	
	<<= >>=	Compound assignment by bitwise left shift and right shift	
	&= ^= =	Compound assignment by bitwise AND, XOR, and OR	
16	,	Comma	Left-to-right

- h. Svi aritmetičke operacije se izvršavaju slijeva udesno, osim operatora dodjeljivanja koji se izvršavaju zdesna ulijevo. Npr, 7+4+8 se izračunava kao (7+4)+8. Redolsjed izvršavanja operacija je kao u matematici: prvo se izvršavaju operacije množenja, dijeljenja i modula, pa tek onda sabiranje i oduzimanje, osim ako zagradama nije drugačije naznačeno. Npr: 7+4*5 ima vrijednost 27, dok izraz ((7+4)*5-3)/2 ima vrijednost 26. U aritmetičkim izrazima koriste se isključivo male zagrade, jer srednje zagrade (tj. [i]) i velike zagrade (tj. { i }) imaju posebno značenje u jeziku C++.
- i. Vodite računa da ako su svi argumenti u izrazu cijeli brojevi, rezultat je također cio broj, čak i ako se koristi dijeljenje. Npr. 13/4 daje rezultat 3, a ne 3.25. Ako želimo tačan rezultat, moramo pisati 13.0/4 ili 13/4.0 ili 13.0/4.0. Obratite pažnju da je rezultat izraza 13/4*2.0 jednak 6.0 a ne 6.5.
3. Zadaci za vježbu. Zadaci označeni simbolom * su, po našem mišljenju, teži od ostalih zadataka.
- a. Napisati program koji učitava cio broj i štampa tekst kao u primjeru. Razmaci, mala i velika slova su važni.

Ulaz	Izlaz
179	The next number for the number 179 is 180. The previous number for the number 179 is 178.

- b. Napisati program koji učitava cio broj m i štampa njegovu cifru stotina i njegovu cifru jedinica u jednom redu, razdvojene jednim blankom.
- c. Učitati tri pozitivna realna broja aa, bb i cc i odrediti da li se može konstruisati trougao čije su stranice dati brojevi. Koristiti promjenljivu tipa bool.
- d. Napisati program koji učitava realne brojeve x i y i izračunava i štampa vrijednost sljedećih izraza: $\frac{x-y}{1+y}$, $\frac{x^3}{3} - 3y^2 + \frac{x+1}{2y+3}$ i $1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4}}}$.
- e. Napisati program koji učitava cio broj d koji predstavlja rastojanje u centimetrima. Odrediti koliko cijelih metara ima u tom rastojanju. Npr. 324cm imaju 3 metra.
- f. Dimenzije pravougaonika su 543 i 130. Napisati program koji štampa koliko kvadrata stranice 65 je moguće izrezati iz tog pravougaonika?
- g. Dat je četvorocifreni prirodan broj. Napisati kod koji štampa zbir kvadrata cifara tog broja.

- h. Napisati program koji učitava trocifreni prirodan broj n i štampa broj koji se dobija zamjenom prve i posljednje cifre.
- i. Dat je četvorocifren broj. Odrediti broj koji se dobija zamjenom treće i druge cifre. Npr. od 5804 dobija se 5084.
- j. Mirkova šifra za sef se predstavlja šestocifrenim prirodnim brojem. Mirko je osmislio šifru na sljedeći način: prvo zamisli šestocifreni prirodan broj $n=abcdef$, pa zatim prvo krene taj broj i na okrenuti broj doda brojeve $m=ace$ i $p= bdf$. Ako je dobijeni broj šestocifren, tada je to šifra. Ako je sedmocifren, tada otkine posljednju cifru sa desne strane i to proglaši za šifru. Pomozite Mirku i napišite program koji izračunava šifru po opisanom postupku. Primjer: Neka je $n=102456$, tada je šifra $654201+125+046= 654372$.
- k. U trocifrenom broju x precrtana je posljednja cifra. Zatim su u dobijenom dvocifrenom broju cifre zamijenile mjesta, pa je na taj dvocifreni broj dopisana na kraj posljednja cifra broja x i dobijen je broj n . Ako je dat broj n , odrediti broj x .
- l. (*) Dat je realan broj a . Koristeći samo operaciju množenja i pomoćne promjenljive izračunati:
- i. a^7 za 4 operacije
 - ii. a^{10} za 4 operacije
 - iii. a^{21} za 6 operacija
 - iv. a^{64} za 6 operacije
 - v. a^3 i a^{10} za 3 operacije
 - vi. a^2 , a^5 i a^{17} za 6 operacija
- m. (*) Date su cifre dva broja: jednog trocifrenog (a_3, a_2 i a_1) i jednog dvocifrenog (b_2 i b_1). Cifre a_1 i b_1 su cifre jedinica, cifre a_2 i b_2 su cifre desetice, a a_3 je cifra stotina. Ako je poznato da je zbir ta dva broja trocifren broj, odrediti cifre zbira.
- n. (*) Dat je cio broj k ($1 \leq k \leq 180$) i niz cifara 10111213...9899 koji se dobija kada se svei dvocifreni brojevi redom zapišu jedna iza drugog. Za dato k , odrediti dvocifreni broj koji sadrži k -tu cifru u datom nizu. Npr., za $k=7$, traženi broj je 13.